WHAT IS CLAIMED IS:

1. A network protocol processor system, the system comprising:

an interface to receive a packet;

a cache to store context data for the packet; and

a processing engine to process the packet using context data in the cache.

2. The system of claim 1, further comprising:

a working register to store the context data for a current connection that is being processed.

3. The system of claim 1, wherein the cache is capable of storing and retrieving context data for multiple connections.

4. The system of claim 1, wherein the interface comprises at least one of a host interface and a network interface.

5. The system of claim 4, wherein the host interface interacts with a doorbell queue, a completion queue, and an exception/event queue.

6. The system of claim 5, wherein each of the doorbell queue, the completion queue, and the exception/event queue is a data structure.

7. The system of claim 6, wherein each data structure has a priority mechanism.

8. The system of claim 5, further comprising:

processing logic to store the packet incoming from the host interface into the doorbell queue;

host memory to store descriptors that are pointed to by the packet;

processing logic to access the descriptors in host memory for storage in the cache; and

a scheduler to perform a hash based table lookup against the cache to correlate the packet with context data, to load the context into the working register when the context data is found in the cache, and to schedule a host memory lookup when the context data is not found in cache.

9. The system of claim 8, further comprising:

a Direct Memory Access (DMA) controller; and

processing logic to notify the DMA controller to transfer data from host memory to the transfer queue.

10. The system of claim 5, wherein the DMA controller is capable of storing data from the header and data queue into host memory.

11. The system of claim 4, wherein the network interface interacts with a header and data queue and a transmit queue.

12. The system of claim 11, further comprising:

processing logic to store the packet incoming from the network interface into the header and data queue;

a working register;

a scheduler to perform a hash based table lookup against the cache to correlate the packet with context data, to load the context into the working register when the context data is found in the cache, and to schedule a host memory lookup when the context data is not found in cache.

13. The system of claim 1, further comprising:

a working register to store data for use by the processing engine; and

a scheduler to locate and load the context data into the working register.

14. The system of claim 1, further comprising:

a Direct Memory Access transfer queue; and

a Direct Memory Access receive queue.

15. The system of claim 1, further comprising:

a timer; and

a hardware assist to translate a virtual address to a physical address.

16 The system of claim 1, further comprising:

a thread cache to store intermediate system state;

a core receive queue;

a working register;

scratch registers;

a pieplined arithmetic logic unit; and

an instruction cache.

17. The system of claim 16, further comprising:

a high bandwidth connection between the thread cache and the working register for parallel transfer of intermediate system state between the thread cache and the working register.

18. The system of claim 1, further comprising:

an instruction cache to store code relevant to specific processing, while remaining instructions are stored in at least one of host memory and cache to store context data.

19. The system of claim 1, further comprising a new instruction set including context access instructions, hashing instructions, multi-threading instructions, Direct Memory Access instructions, timer instructions, and network to host byte order instructions.

20. The system of claim 1, further comprising:

a scheduler coupled to the cache;

a working register coupled to the cache; and

processing logic in the processing engine to store context data in the working register into the storage area when processing of the packet has stalled.

21. The system of claim 20, wherein the packet is a first packet and further comprising:

processing logic to load context data for a second packet from the storage area into the working register.

22. The system of claim 21, further comprising:

processing logic to restore the context data for the packet into the working register.

23. The system of claim 1, further comprising:

a Direct Memory Access (DMA) controller coupled to the processing logic, wherein the DMA controller is capable of transferring data independently while the processing engine continues context processing in parallel.

24. A network protocol processor system, the system comprising:

a first interface to receive a first packet, wherein the first interface is coupled to a source of a first clock signal having a first frequency;

a second interface to receive a second packet, wherein the second interface is coupled to a source of the first clock signal having the first frequency;

processing logic to process the first packet and the second packet, wherein at least one component of the processing logic is coupled to a second clock signal having a second frequency different than the first frequency.

25. The system of claim 24, wherein the second frequency is higher than the first frequency.

26. The system of claim 24, further comprising:

a computing device coupled to the first interface; and

a network interface controller coupled to the second interface.

27. The system of claim 24, further comprising:

a storage area to store context data for at least one packet;

a subset of the storage area to store context data for one packet; and

processing logic to retrieve context data for the first packet, wherein the subset of the storage area is coupled to the processing logic.

28. A method for processing a packet, comprising:

receiving a packet;

locating context data for the packet in a storage area; and

processing the packet using the context data.

29. The method of claim 28, further comprising:

performing a lookup against the storage area to correlate the packet with context data;

loading the context into a working register in response to locating the context data in the storage area; and

scheduling a lookup of context data in response to determining that the context data is not in the storage area.

30. The method of claim 29, further comprising:

storing context data from the working register into the storage area when processing of the packet has stalled.

31. The method of claim 29, further comprising:

updating process results to the working register;

updating the storage area with the results in the working register; and

updating a thread area with the results in the working register.

32. The method of claim 31, wherein the packet is a first packet and further comprising:

loading context data for a second packet from the storage area into the working register.

33. The method of claim 32, further comprising:.

restoring the context data for the first packet into the working register.

34. An article of manufacture comprising a storage medium having stored therein instructions that when executed by a computing device results in the following:

receiving a packet;

locating context data for the packet in a storage area; and

processing the packet using the context data.

35. The article of manufacture of claim 34, wherein the instructions when executed further result in the following:

performing a lookup against the storage area to correlate the packet with context data;

loading the context into a working register in response to locating the context data in the storage area; and

scheduling a lookup of context data in response to determining that the context data is not in the storage area.

36. The article of manufacture of claim 35, wherein the instructions when executed further result in the following:

storing context data from the working register into the storage area when processing of the packet has stalled.

37. The article of manufacture of claim 36, wherein the instructions when executed further result in the following:

updating process results to the working register;

updating the storage area with the results in the working register; and

updating a thread area with the results in the working register.

38. The article of manufacture of claim 37, wherein the packet is a first packet and wherein the instructions when executed further result in the following:

loading context data for a second packet from the storage area into the working register.

39. The article of manufacture of claim 38, wherein the instructions when executed further result in the following:

restoring the context data for the first packet into the working register.